LGURJCSIT
Volume No. 1, Issue No. 1 (Jan-March 2017) pp. 1-12

Algorithm for Coding Person's Names in large Databases / Data Warehouses to Enhance Processing Speed, Efficiency and Reduce Storage Requirements

Aftab Ahmad Malik

Abstract-: A technique to codify the names of people in Databases and Data warehouses to reduce the storage requirements and enhance processing speed is presented. It is estimated that the storage requirement can be reduced, which is normally dominated by the entries of first name, middle name and last name and storage requirements for Data warehouses keep on increasing. The scheme of the Algorithm replaces the first name, middle name and last name in numeric charterers instead of alpha characters for storage. It decodes them into actual Alpha Characters at the time retrieval. The step by step analysis and working of the algorithm is presented.

Keywords: - Alpha-characters, Coding, decoding, names, numeric characters

1. INTRODUCTION

It is normal practice to assign codes to countries, cities, airports, flights, trains, organizations, departments, roads, houses, streets, students roll numbers, and books like ISBN, research journals, publishers and many other objects in daily life. For example, all States of US and all cities of U[1]K have unique codes. Moreover, income tax number, passport, Customer identity numbers and international telephone dialing codes are extensively used in Database Management applications.

[1] Discusses name entity problem and evaluation and [3] presents Layered space-time codes have been designed to use with a decoding algorithm based on QR decomposition. [4], [6] provide description of Oracle decode key word and decode function, which codes and decodes the relevant field entries of a database, effectively. The key word decodes works on "if-then" logic and provides result based on the based on the value of a column or function. According to [4][6] and Oracle Complete Reference Series the syntax is as follows:

Select...DECODE (Value, If1, Then1, [If 2, Then 2, ...,] Else) ...From ...;

Example: Select Distinct City, DECODE (City, 'London', '111', 'New York', '367', 'Chicago', '982', City) AS code from Cities;
The 'City' in the first argument represents the column name.
Output:

Prof. Aftab Ahmad Malik
Department of Computer Science
Lahore Garrison University
Lahore, Pakistan
Authors email dr_aftab_malik@yahoo.com

```
City        code
------------ -------
London      111
New York   367
Chicago     982
```

Example; SELECT name_id, Decode (idcard_id, 111,'fname, 347, mname, 789,'lame') from name_code.

Further, [2] Exhibits the working of decode function as follows:

Decode (expression, search, result [, search, result] …..[, default])

The Syntax of Oracle /PLSQL decode function is simple and performs its working in an IF-then –else statement which works with Oracle versions 9i,10g,11g and 12 c.

[5] Proposes a decoder which is based on the min-sum decoder for binary LDPC codes called "Extended min-sum (EMS)".

[8] And [9] describe Pakistan's National Database & Registration Authority (NADRA), equipped with one of largest fully integrated Data warehouse of the world. It facilitates issuance National Identity cards and passports for citizens apart from authentication and verification of information regarding facial images and fingerprints. According to [8], it is equipped with World Largest Facial Library of 47 million images. This Data warehouse [8] is very useful having 90 Million citizen registered including 30 million children; more than 56 million id-cards issued to citizens. According to [9], its processing speed 18 trillion Instructions per sec and storage capacity is 60 Terabyte. [10] Suggests the formation of code –decode tables, but there are a few disadvantages such as we are unable to enforce referential in Tegrity (foreign Key).

Definition: The number of (machine) instructions which a program executes during its running time is called its time-complexity, while the number of memory cell required in an Algorithm is Space Complexity.

Better Time Complexity ensures the faster running of Algorithms. On the other hand, a good Algorithm must have small Space Complexity. It is well established fact that the Time Complexity and Space Complexity of an Algorithm have a trade-off and though are different aspects of determining the efficiency of an Algorithm. Certainly time Complexity changes with a change in size FIRST NAME (326) MIDDLE NAME (111) and LAST NAME (231) of the input.

2. PROCEDURE

Most of the Databases and Data warehouses are dominated with the information about First Name, middle name, Last Name, Father's name (First Name, middle name, Last Name) Address, Email, Sex, State, City, Country, Phone Number etc.

The person's name normally requires at least 40 Characters. When names are combined with Father's Name the Character length required for the both names become 80 Characters. Off course some names are larger than 40-character length.

Name: 40 Alpha Characters

FIRST NAME	MIDDLE NAME	LAST NAME

Father's Name: 40 Alpha Characters

FIRST NAME	MIDDLE NAME	LAST NAME

We propose to code and decode name and Father's name into numeric Characters as shown below:

Name: 3+3+3=9 Numeric Characters

FIRST NAME	MIDDLE NAME	LAST NAME
234	678	231

Father's Name: 3+3+3=9 Numeric Characters

2.1 Coding of Names

It is usual practice that where the human data is stored into computer's memory, forty characters are reserved for one name. In this way one individual name and his/her father's name, we need eighty characters. For storing the information of a large number of persons working in a particular organization the large amount of computer memory is required. Let us now estimate of our requirements in both modes of storing names i.e., directly in alphabetical characters and under digital coding system.

Requirement of storage for name and Father's Name in Alpha Characters:80

Requirement of storage for name and Father's Name in Numeric Characters: 09. Therefore, with the implementation of Algorithm presented in this paper a remarkable reduction in storage occurs, particularly when the number of records in database is in millions. For the purpose of the coding and decoding strategy we have divided one name into three parts:

i. First Name
ii. Middle Name
iii. Last Name,

2.2 How our Algorithm works:

Supposing that each part of the name consists of about 12 to 14 characters. We have collected all possible names from the telephone directory and allocated codes of three digits to them by the program NAMETAB. When the program NAMETAB is executed, the computer is ready to ACCEPT name from the screen while the message "ENTER NAME:" is also displayed on the screen. While entering the names, the computer allocates a code of three digits starting from 001 to each name. All the names collected from telephone directory have been entered and allocated codes initially.

For example, the name "MUHAMMAD AKRAM BUTT" has three parts:

First: MUHAMMAD
Second: AKRAM

Third: BUTT

According to the program built up two files NAME Code File (NAME—CDFL) and NAME Decode File (NAME-DCODE) have been maintained. In order to keep the track of the last code allocated to name, last code is saved on a Unit Record File (LAST-NUMBR). Since we have allocated codes of three digits to each part of the name, therefore, nine digits are required for one name. Since two digits are stored into one byte of memory and in this way the required bytes for storing these digits are more efficient resulting in Net % Saving of 88.18% saving in computer memory using our coding-decoding system.

Initially we have Name Code File and Name Decode File. Our Algorithm NAMETAB, which initializes the Name Code File (NAME—CDFL) and Name Decode File (NAME-DCODE). Now we also explain the algorithm used for coding and decoding names and inclusion exclusion of names from these files. First of all, we initialize the Name Code File and Name Decode File by executing the program NAMETAB. A name is accepted in working storage and checked whether it is alphabetic or not. If it is not alphabetic, the message "NAME IS NOT ALPHABETIC" is displayed upon the screen and no code is generated for this name. The computer now is ready to ACCEPT another name. If the name is alphabetic, the name is placed in the name fields of Name Code File and Name Decode File. Initially the code is set equal to zero. Now add 1 to code and move this code to the name-code fields of both the files and the records are written on the files. If the name already exists in the files, the message "DUPLICATE NAME FOUND" is displayed and 1 is subtracted from the code since 1 has been added before. In this way a sufficiently large number of names have been entered in the Name Code File and Name Decode File. In the end we enter blank to come out of the loop of accepting name and allocating code. The last code generated is moved to the LAST-CODE field of the Unit Record File LAST-NUMBR and the record is written into the file.

2.2.1 Coding Algorithm:

The following algorithm is used to allot codes to names:

```
CODIFY-NAMES:

            MOVE FNAME1 TO NAME OF CODE-REC.
            PERFORM CODE-PARA THRU EXIT-POINT.
            MOVE NAME—CODE TO W—FNAME.
             IF MNAME1 = SPACE
            MOVE ZERO TO W-FNAME
             ELSE ____________________

            ____________________
            CODE-PARA;
            IF NAME IS NOT ALPHABETIC

            _________________________

            _________________________
            GENRT-CODE.
```

```
                ADD 1 TO LAST-CODE
               ___________________
         EXIT-POINT.
        EXIT.
```

The first name (FNAME1) is moved to the name field of the CODE-REC (The record name of the Name Code File), since Name is used to access this file. First of all, this name is checked whether this is alphabetic or not because the Name Code File has only alphabetic names. If the name is not alphabetic, the message "NAME IS NOT ALPHABETIC" is displayed upon the screen and the algorithm is terminated. If the name is alphabetic, an attempt is made to read the Name Code File. If reading is successful, it means that this particular name exists in the Name Code File and the corresponding Name-code is moved to the specified space reserved for the code of First Name in the Master Record. If reading is unsuccessful, it means this particular name whose code is required does not exist in the Name Code File and the control transfers to GENRT-CODE para of the program to generate a new code for this particular name. For this purpose, the Unit Record File LAST-NUMBR is read to get the last code allotted, 1 is added to the last code field LAST-CODE. The last code and the particular name are moved to Name-Code field and Name field respectively of both the records CODE-REC and DECODE-REC of Name Code File and Name Decode File. The records are written into the files to update the files for later use. Since the names of some persons consist of only one part for example, ABDULLAH, the middle name (MNAME1) is first checked whether it is a blank or not. If it is a blank, zero is moved to the specified space for code of middle name. If it is not a blank, the same procedure of checking alphabetic, reading Name Code File and moving the Name-Code to the specified space is carried out.

If certain names consist of only two parts, for example MUHAMMAD ABDULLAH the last name (LNAME1) is first checked whether it is a blank or not. Again if it is blank, zero is moved to the specified space for code of last name. If it is not a blank, the same procedure as for middle name is carried out. In the end the record L-NO-REC of the Unit Record File is rewritten to update the file. In this way the coding of name is complete and all the aspects of the coding algorithm have been discussed.

Decoding Algorithm: In this section we state and describe the algorithm which is used for umer decoding names. The following algorithm is used to decode the names:

```
DECODE-ROUTINE.
                MOVE W-FNAME TO ANAME-CODE OF DECODE-REC
                MOVE MNAME TO FNAME1
                IF W-MNAME = ZERO
                        MOVE SPACES TO MNAME1
                ELSE
               _________________
               _________________
               READ -NAME-DECODE.
```

```
                READ NAME—DECODE
          INVALID KEY DISPLAY "SORRY UNABLE TO READ".
```

The First Name Code (W-FNAME) is moved to the name code field (ANAME-CODE) of the DECODE-REC (the record name of the Name Decode File), since the ANAME-CODE is used to access this file. An attempt is made to read the Name Decode File (NAME-DECODE). If the reading is not successful, the message "SORRY UNABLE TO READ" is displayed and the algorithm is terminated. If the reading is successful, the corresponding name is moved to the specified space for decoded name. Now, the middle name code (W-MNAME) is checked whether it is zero or not. If it is zero, blanks are moved to the specified space for decoded name. If it is not zero, the same procedure of reading the Name Decode File and moving the name to the specified space is carried out. The last name code (W-LNAME) is also checked whether it is zero or not. If it is zero, blanks are moved to the specified space for decoded name. If it is not zero, same procedure as for middle name code is carried out. Now the complete decoded name is ready for output purpose.

Exclusion of Names from Name Code File & Name Decode File:

In this section we describe the method for excluding the names from Name Code File (NAME—CDFL) and Name Decode File (NAME-DCODE). If the name is not alphabetic, it is detected by the algorithm CODIFY-NAMES discussed in the previous section and no code is allotted to that name and the algorithm terminates. On the other hand, if unintentionally wrong spellings of a name are entered, this time the name is alphabetic, a code is allotted to this name and a useless record is maintained in both the files. Moreover, an unnecessary name code is generated. To get rid of this record, we have a separate program DELETE-NAME, which is executed as and when needed. We delete the name and its code from both the files in order to save storage and avoid from occupying storage for meaningless names. When the execution of the program DELETE-NAME starts, the following message is displayed upon the screen after erasing it.

SCREEN LAYOUT:

```
          "ENTER NAME ************
                         WHICH IS TO BE DELETED FROM
                        NAME CODE FILE . . . . . . . . .... NAME-CDFL"
```

The computer now is ready to ACCEPT the name and the cursor is on the first asterisk displayed on the first line. The name is accepted in the Name field (NAME) of the Name Code File and an attempt is made to delete the record from the file. If the record with this particular name exists in the file, it is deleted and the message "RECORD DELETED

PRESS NEW LINE" is displayed upon the screen and on pressing the key labelled "NEW LINE", the same procedure of accepting and deleting name is carried out.'0n the other hand if the record does not exist for the entered name, the message "RECORD NOT FOUND" is displayed and after pressing the key "NEW LINE" the control is again at the start of the loop. Each time a name is accepted, it is compared with the letter "A" because we have used this letter for terminating the loop, the moment it equals, and the loop is terminated and the control enters into another loop which is used to delete name and its code from the Name Decode File. For this file, as discussed in the section 4.2, the field by which it is accessed is ANAME—CODE. In the start of the loop the following message is displayed upon the screen after erasing it:

SCREEN LAYOUT

*"ENTER ANAME-CODE ****
WHICH IS TO BE DELETED FROM
NAME DECODE FILE . . . . . NAME-DCODE"

The computer now is ready to ACCEPT the code and the cursor is on the first asterisk displayed on the first line. The code is accepted in the Name Code field (ANAME-CODE) of the Name Decode File and an attempt is made to delete the record from the file. If attempt is successful, the message "RECORD DELETED' PRESS NEW LINE" is displayed and on pressing the key "NEW LINE", the same procedure of accepting and deleting the record is carried out. On the other hand, if the attempt of deleting results unsuccessful, the message "RECORD NOT FOUND "PRESS NEW LINE" is displayed and after pressing the key "NEW LINE" the control goes to the start of the loop. Each time a Name-Code is accepted, it is compared with the value of a figurative constant ZERO because we have used zero to terminate the loop, the moment it equals, the control comes out of the loop. The Unit Record Pile LAST—-NUMBR is updated by entering the last code from terminal for later use. Now the computer asks whether the complete list of names and their codes is required or not. If we enter "N" the program is terminated without giving the list. On the other hand, if we enter "Y", the Name Code File (NAME-CDFL) is first made ready for sequential reading with the START statement, since it is an indexed file, and a complete list of names and their codes is prepared by repeatedly reading the Name Code File sequentially.

3. CONCLUSION

The purpose of this research is to reduce the storage requirement of a large Data warehouses like NADRA. In case the scheme is implemented the speed for on line transactions will be enhance with reduction in 80% of storage requirement.

4. ACKNOWLEDGMENT

The author wishes to thank Chancellor, Dean and Engineer Mujtaba Asad, Department of Computer Sciences Lahore Garrison University, Lahore.

5. REFERENCES

[1]: Danial M bikel, Richerd Schwartz, Ralph M. Weiischedel," An Algorithm that learns what is in a name", Machine Learning 34,211-231(1999)

[2]: Oracle decode Syntax
https://www.docs.oracle.com/cd/B19306_01/server.102/b14200/functions040.htm

[3]: Wübben, D; Böhnke, R; Rinas, J; Kühn, V; Kammeyer, K D. "Efficient algorithm for decoding layered space-time codes", Electronics Letters 37.22 (Oct 25, 2001): 1-2

[4] Oracle decode Function: http://www.oradev.com/decode.html

[5]: David Declercq, ETIS ENSEA," Decoding Algorithms for Nonbinary LDPC Codes Over GF(q)", IEEE Transactions on Communic...Volume: 55 Issue: 4

http://www.ieeexplore.ieee.org/document/4155118/

[6]: Database SQL Reference; Oracle Database Online Documentation, 10g Release 2 (10.2) / Administration
https://www.docs.oracle.com/cd/B19306_01/server.102/b14200/functions040.htm

[7]: Mene Mene, Tekel:" SQL an,d its Sequels,Muising on MySQL", http://www.ocelot.ca/blog/blog/2013/09/16/representing-sex-in-databases/

[8]: Nadra: National Database and Registration Authority, Pakistan. https://www.nadra.gov.pk/

[9]: Asim Sardar "Case Study: Information System Reforms for improving Governance ",
https://www.siteresources.worldbank.org/PSGLP/Resources/InformationSystemsReformsforImp rovingGovernanace

[10]: Vincent Rainardi Data Warehouse and Business Intelligence,

https://dwbi1.wordpress.com/2015/04/23/code-decode-table/ 23 April 2015